REMARKS

Favorable reconsideration of this application is respectfully requested in view of the amendments above and the following remarks. Claims 1, 3-27, 29 and 31 are pending, of which claims 1, 18 and 25 are independent.

Claims 1, 3-27, 29, and 31 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Sakai (Sakai, US 2004/0103410 A1) in view of Gold et al. (Gold, US 2002/0184473 A1).

These rejections are respectfully traversed for at least the reasons set forth below.

Claims 1, 3-27, 29, and 31 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Sakai in view of Gold.

I. Finality must be withdrawn because the rejection of at least claim 25, which has not been amended, is improper for the reasons stated below.

Claim 1 recites,

> A method of allocating registers when compiling source code, said method comprising steps of:
> ...during compiling of the source code, selecting at least one subclass of said selected class of real registers, wherein said at least one subclass includes a register to store said operand.

Neither Sakai nor Gold singly or in combination teach or suggest that when compiling source code selecting at least one subclass of said selected class of real registers to store said operand. The rejection of claim 1 states that Sakai does not expressly disclose selecting a

subclass of the selected class of registers. However, the rejection combines Gold with Sakai to allegedly teach this feature.

Gold fails to teach or suggest selecting a subclass of a selected class during compiling. Gold discloses an instruction scheduler 10 performs register mapping at runtime when executing instructions. See paragraph 27. Thus, Gold fails to teach or suggest steps performed during compiling.

Also, Gold discloses mapping virtual registers to physical registers after compiling because microprocessors operable to execute multiple instructions in a single cycle may execute instructions non-sequentially. A register mapping is performed during execution of instructions to accommodate executing instructions non-sequentially. See paragraphs 4, 5, and 26. Sakai, on the other hand, discloses performing a parallelization process during compiling that generates a target program 3 from a source program 1 for a parallel processing architecture. The parallelization process determines where the intermediate code can be parallelized and reorders instructions to accommodate the parallelization. See paragraph 84. Thus, there is no need to use the register mapping or any of the processes of Gold in Sakai because the target program 3 of Sakai is already optimized for parallel processing. Thus, there would be no reason to combine Gold with Sakai and the motivation to combine is improper.

II. Independent claim 18 has been amended to include the features of claim 19 and now recites,

during compiling of the source code, allocating a plurality of real registers to store a plurality of operands from said intermediate code while generating the intermediate code, wherein the allocating further comprises

determining a type of operand for at least one of said plurality of operands;

allocating a location in memory for the at least one operand in response to said operand being a particular type of operand; and

allocating a real register for said operand.

Neither Gold not Sakai teach or suggest performing these steps during compiling.

III. Independent claim 25 recites a compiler comprising a register allocation stage, an optimization stage and a final code stage generating machine readable code. The register allocation stage is configured to select a class of registers and select a subclass of said class of registers and allocate a real register from said selected subclass of registers for one of said plurality of operands, said one operand being of a particular type of operand.

Gold was combined with Sakai to teach selecting a subclass of registers. Claim 25, however, recites an optimization stage in a compiler configured to select a subclass of said class of registers and allocate a real register from said selected subclass of registers for one of said plurality of operands. Gold fails to teach a compiler selecting a subclass of registers. Instead, Gold performs register mapping at runtime and not during compiling. Also, it would not have been obvious to combine Gold with Sakai for the reasons stated with respect to claim 1. Thus, claims 18-27, 29, and 31 are believed to be allowable.

Dated: October 9, 2005

By _____

Ashok K. Mannava
Registration No.: 45,301

MANNAVA & KANG, P.C.
8221 Old Courthouse Road
Suite 104
Vienna, VA 22182
(703) 652-3822
(703) 880-5270 (facsimile)